# AUTOMATIC PHASE AND FREQUENCY ADJUSTMENT CIRCUIT AND METHOD

This application claims priority to U.S. Provisional Patent Application Serial No. 60/213,202 filed June 21, 2000, which is incorporated by reference in its entirety.

## BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a system adapted to visually display image data and, more particularly, to a circuit and method adapted to automatically adjust the phase and frequency of a pixel clock derived from analog image data.

2. Description of the Related Art

Computers, televisions, video cameras, and other sources generate analog image data that is often digitized for eventual display on a digital display device. A pixel clock is necessary to properly digitize the analog image data. The pixel clock clocks the analog to digital converter that converts the analog image data into digital image data. Although the analog image data does not include the pixel clock, it does include a horizontal synchronization signal often used to generate the necessary pixel clock.

Figure 1 is a block diagram of a circuit 100 including a phase locked loop (PLL) circuit 112 used to recover a pixel clock 128 from a reference or horizontal synchronization signal 110 for corresponding analog image data 132. The PLL circuit 112 includes a phase detector 114 serially connected to a loop filter 116 and a voltage controlled oscillator (VCO) 118. The PLL feedback loop comprises a programmable divider 120 that receives the VCO clock signal 122 and provides feedback signal 124 to the phase detector 114 responsive to a frequency adjust signal 136. The PLL circuit 112 is well known to those skilled in the art and will not be explained in further detail.

A phase adjust circuit 126 receives the VCO clock signal 122 and provides the pixel clock 128 to the analog to digital converter (ADC) 130 responsive to a phase adjust signal 138. The ADC 130 converts the analog image data 132 into digital image data 134 responsive to the pixel clock 128.

A digital data analysis circuit 140 generates the frequency and phase adjust signals 136 and 138, respectively, from an analysis of the digital data 134. To obtain a noise free

image, the digital data 134 must be properly clocked. That is, the pixel clock 128 must have a phase and frequency appropriate for the analog data 132. The phase adjust circuit 126 uses the phase signal 138 to adjust the phase of the pixel clock 128 such that pixels sampled when the analog data 132 is stable and not in transition. The PLL circuit 112 uses the frequency signal 136 to adjust the frequency of the VCO clock 122 and consequently, the pixel clock 128. That is, such that the PLL circuit 112 is set to the proper multiple of the reference or horizontal synchronization signal 110.

The phase and frequency adjust circuit 100 relies on a feedback loop that examines the digital data 134 and makes adjustments to the phase and frequency of the pixel clock 128 until the digital data 134 is stable and noise free. One disadvantage associated with circuit 100 is that the phase and frequency adjustment might take several seconds to complete. Another disadvantage is that the error rate (or effectiveness) depends on the digital data 134. Yet another disadvantage is that the circuit 100 might not work correctly or consistently for images that do not completely fill the digital display device (not shown) at the expected resolution.

Accordingly, a need remains for an improved automatic phase and frequency adjust circuit and method.

## BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features, and advantages of the invention will become more readily apparent from the following detailed description of a preferred embodiment that proceeds with reference to the following drawings.

Figure 1 is a block diagram of a phase and frequency adjust circuit.

Figure 2 is a block diagram of an embodiment of an automatic phase and frequency adjust circuit of the present invention.

Figure 3 is a timing diagram of the signals associated with an embodiment of the edge detector circuit shown in Figure 2.

Figure 4 is a timing diagram of the signals associated with an embodiment of the phase adjust circuit shown in Figure 2.

Figure 5 is a block diagram of embodiments of the phase hit detector and phase counter circuits shown in Figure 2.

Figure 6 is a timing diagram of the signals associated with embodiments of the phase hit detector and phase counter circuits shown in Figures 2 and 5.

Figure 7 is a flowchart of a method for automatically adjusting the phase and frequency of image data according to an embodiment of the present invention.

Figure 8 is a flowchart of a method for automatically adjusting the frequency of the image data according to an embodiment of the present invention.

Figure 9 is a flowchart of a method for automatically adjusting the phase of the image data according to an embodiment of the present invention.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to Fig. 2, an automatic phase and frequency adjust circuit 200 includes a low-jitter PLL circuit 212 used to recover a PLL clock 222 from a reference signal 210. The reference signal 210 in one embodiment is a horizontal synchronization signal corresponding to analog image data 232. The PLL circuit 212 includes a phase detector 214 serially connected to a loop filter 216 and a VCO 218. The phase detector 214 produces a direct current or low frequency signal proportional to the phase difference between the incoming signal and the signal 224. The VCO 218 is an oscillator whose frequency is proportional to an externally applied voltage. When the PLL 212 is locked on to the reference signal 210, the VCO 218 frequency is exactly equal to that of the reference signal 110.

The PLL feedback loop comprises the VCO 218 and a programmable divider 220 that receives the PLL clock 222 and provides feedback signal 224 to the phase detector 214 responsive to a frequency adjust signal 236.

If, for example, the frequency of the reference signal 210 shifts slightly, the phase difference between the PLL clock 222 and the reference signal 210 will begin to increase with time. This will change the control or input voltage 217 on the VCO 218 in such a way as to bring the VCO frequency back to the same value as the reference signal 210. Thus, the PLL circuit 212 can maintain lock when the input frequency changes and the VCO input voltage 217 is proportional to the frequency of the reference signal 210. The operation of the PLL circuit 212 is well known to those skilled in the art and will not be explained in further detail.

An edge detector circuit 240 generates an edge pulse signal 254 of arbitrary width corresponding to a transition of the analog signal 232 above a predetermined threshold (not shown). The threshold (not shown) allows the circuit 200 to ignore small amounts of noise on the analog data signal 232. The threshold (not shown) might be fixed or user programmable through external and/or internal means (not shown) such as system software (not shown). The threshold, for example, might be set at ½ of a full signal range or it might

be dynamically set before, during, or both before and during reception of the analog data signal 232 depending on a variety of circumstances such as the type of analog data signal received. The edge detector 240, for example, might detect an analog data signal transition above ½ of the full signal range and will generate a corresponding edge signal pulse 254. A person skilled in the art should recognize that the threshold might be set in a variety of ways and implemented in a variety of manners.

Figure 3 shows exemplary timing signals for an analog signal 332 and an edge pulse signal 354. Referring to Figures 2 and 3, the edge detector 240 (Figure 2) generates a pulse 312 on the edge pulse signal 354 responsive to detecting a transition 304 above a predetermined threshold, e.g., ½ signal range. Similarly, the edge detector 240 (Figure 2) generates pulses 314, 316, and 318 responsive to detecting transitions 306, 308, or 310 above a predetermined threshold.

The edge detector 240 might detect rising, falling, or rising and falling edge transitions of the analog signal 232. For example, the edge detector 240 (Figure 2) detects rising edges 304, 306, and 310 above a predetermined threshold and thereby generates corresponding pulses 312, 314, and 318 on the edge pulse signal 354. Similarly, the edge detector 240 (Figure 2) detects a falling edge 308 above a predetermined threshold and thereby generates pulse 316 on the edge pulse signal 354.

Returning to Figure 2, a phase adjust circuit 245 receives the PLL clock 222 and generates a pixel clock 228 responsive to a phase adjust signal 238. The phase adjust circuit 245 comprises a clock delay circuit 244 connected to a multiplexer circuit 246. The clock delay circuit 244 generates a plurality of clock phases 260 by delaying the PLL clock 222.

An embodiment of the clock delay circuit 244 is an n-stage delay locked loop (DLL). The n-stage DLL circuit 244 generates n different clock phases 260 that are each 360/n degrees out of phase with the reference signal 210. Figure 4 shows timing signals associated with an 8-stage (n=8) DLL circuit 244 comprising phase signals ph0-ph7 that are 45 degrees out of phase.

The multiplexer circuit 246 selects one of the phases 260 responsive to the phase adjust signal 238. In the case of an 8-stage DLL circuit 244, the multiplexer circuit 246 comprises an 8 to 1 multiplexer controlled by a three bit phase adjust signal 238.

An alternative embodiment of the phase adjust circuit 245 is a phase adjust circuit 245 that delays the reference signal 210 at the input of the phase detector 214. That is, the phase

adjust circuit 245 is positioned at an input to the PLL circuit 212 and delays the reference signal 210.

A phase detector circuit 247 identifies one of the clock phases 260 most closely associated with the transition of the analog data signal 232 and generates the phase and frequency adjust signals 238 and 236, respectively. The phase detector circuit 247 substantially simultaneously analyzes the clock phases 260, generates a corresponding plurality of phase hit enable signals 249, and asserts the phase hit enable signal 249 corresponding to the transition of the analog signal 232. The phase detector circuit 247 does this analysis every clock cycle if the analog data signal 232 is transitioning every clock cycle. A person skilled in the art should understand that the phase detector circuit 247 might operate at a variety of different speeds corresponding to the analog data signal 232.

The phase hit counter 250 receives the plurality of phase hit enable signals 249 and generates a corresponding plurality of phase count signals ph0cnt...phncnt by counting the number of times corresponding phase hit enable signals 249 are asserted. The phase count analysis circuit 252 receives the phase count signals ph0cnt...phncnt and generates the phase and frequency adjust signals 238 and 236, respectively. A person skilled in the art should recognize that the phase count analysis circuit 252 might be implemented in hardware, software, or a combination of both. The phase hit counter 250 additionally receives an enable and a reset signal for enabling and resetting the counting.

Figure 5 is a block diagram of an embodiment of the phase detector circuit 247 shown in Figure 2. Referring to Figure 5, the phase detector circuit 547 includes a plurality of flip flops, e.g., D-flip flops 502-512. The flip flops, e.g., D-flip flops 502-512, receive the edge pulse signal 554 at the data input. A corresponding one of the clock phase signals clock each of the flip flops. For example, flip flop 502 receives the edge pulse signal 554 at its data input and clock phase signal ph0 at its clock input and flip flop 510 receives the edge pulse signal 554 at its data input and clock phase ph7 at its clock input.

The phase detector circuit 547 double clocks the edge pulse signal 554 to avoid metastability issues. That is, two levels of flip flops clock the edge pulse signal 554 with a corresponding clock phase signal. For example, flip flops 502 and 504 clock the edge pulse signal 554 with clock phase signal ph0. Thereafter, the edge pulse signal 554 is realigned such that all the results from the different clock phases of a given clock cycle are available at the same time. With this realignment, the phase hit detectors 248 and phase hit counters 250

run synchronously off the same phase of the clock e.g., phase 0. This realignment is optional but convenient.

The clock phase signal ph0 is used twice: once to look for edges between clock phase signals ph0 and ph1 and again to look for edges between clock phase signals ph7 and ph0.

The phase hit detector 518 looks for adjacent phase out signals ph0_out…phn_out where one is at a first level, e.g. 0, and the next is at a second level, e.g. 1. The phase hit detector 518 then asserts the corresponding phase hit enable signal, e.g., ph0_hit_en…phn_hit_en for one clock cycle. The phase hit detector 518 asserts the corresponding phase hit enable signal according to the following equation:

ph_n_hit_en = ph_n_out * !ph_n-1_out.

The phase hit enable signals, e.g., ph0_hit_en…phn_hit_en, advance the corresponding phase hit counters, e.g., phase hit counters 520-524. The phase hit counters accumulate the number of phase hits on each phase over some counting interval, e.g., a predetermined number of scan lines of the image, a selected number of pixels in an image frame, an entire image frame, multiple frames, and the like.

The phase hit counters, e.g., phase hit counters 520-524, might include a count enable signal 256 (Figure 2) and a clear enable signal 258 (Figure 2) to enable and clear, respectively, the count in the phase hit counters.

Referring to Figure 2, a phase count analysis circuit 252 reads the phase hit count signals, e.g., signals ph0cnt…phncnt, and generates the phase and frequency adjust signals 238 and 236, respectively, relative to the analog data signal 232 using the following algorithms.

To determine the correct clock frequency and thereby generate the frequency adjust signal 236 (Figure 2), the phase hit counters 250 count the number of phase hit enable signals ph0_hit_en…phn_hit_en asserted over a predetermined period, e.g., a number of scan lines of the image. The phase count analysis circuit 252 generates the frequency adjust signal depending on an examination of the contents of the phase hit counters, e.g., phase hit counters 520-524. For example, if only a some of the phase hit counters e.g., less than 50%, show a significant number of hits, then the frequency of the pixel clock 228 is set correctly and the frequency adjust signal 236 remains unchanged. That is, the frequency of the pixel clock 228 is set correctly if only a predetermined low number of the phase hit counters have

hits. Otherwise, the phase count analysis circuit 252 generates a corresponding frequency adjust signal 236.

If, on the other hand, all of the phase hit counters 250, that is 100% of the phase hit counters, have a predetermined high number of hits, the frequency of the pixel clock 228 is set incorrectly and must be adjusted. A predetermined high number of hits indicate that the image data is moving in and out of phase with the PLL clock 222. The circuit 200 adjusts the frequency of the pixel clock 228 by generating a appropriate frequency adjust signal, clearing the counters by asserting the clear signal 258, enabling the counters by asserting the enable signal 256, and reading the phase hit counters again after a predetermined time lapse. The circuit 200 repeats this process until the frequency of the pixel clock 228 is set correctly. Searching the proper frequency can be done by a linear search, binary search, or any other searching method known to a person of skill in the art.

The process described above might be performed several times in one vertical synchronization period since the PLL circuit 212 settles relatively quickly —usually in a few lines e.g., less than 100— after changing the frequency adjust signal 236. The process of measuring the frequency is also relatively quick —usually a few lines. Being able to do multiple iterations during each vertical synchronization period allows the correct frequency to be found quickly.

Once the pixel clock 228 is set to the appropriate frequency, the circuit 200 might adjust its clock phase as follows. The circuit 200 clears the phase hit counters by asserting the clear signal 258 and enables the phase hit counters for a predetermined amount of time by asserting the enable signal 256. The phase count analysis circuit 252 reads the phase hit counters 250. At this point, only a few of the phase hit counters will show any appreciable number of hits (more than one due to the PLL clock jitter). The phase count analysis circuit 252 generates a phase adjust signal 238 corresponding to the phase of the PLL clock associated with the largest number of hits. One advantage is that the circuit 200 might adjust the phase of the pixel clock 228 immediately after only one read of the phase hit counters 250 improving operating speed. Previous techniques for determining proper clock phase typically require changing the phase multiple times before making a determination about which phase is correct. Previous techniques are therefore slow by comparison.

An exemplary operation of the automatic phase and frequency adjust circuit 200 (Figure 2) and the phase detector circuit 547 (Figure 5) will be explained with reference to Figure 6. Referring to Figures 2, 5, and 6, the input data 632 transitions from a first level to a

second level at a time t1 during clock cycle 1. The edge detector 240 (Figure 2) detects the transition of the analog signal 632 and generates a pulse 602. The phase detector circuit 547 identifies and asserts phase out signal ph5_out and ph6_out responsive to the edge pulse signal 654 and the clock phase signals ph0...ph7. That is, the phase detector circuit 547 identifies the phase signal ph5 as most closely associated with the transition of the analog data signal 632 as indicated by the edge pulse signal 654. The phase hit detector 518 looks for adjacent phase out signals ph0_out...ph7_out where one signal is at a first level, e.g., 0, and a second is a second level, e.g., 1. As a result, the phase hit detector 518 enables the phase hit enable signal ph5_hit_en corresponding to the phase signal (ph5) closest to the first transition of the analog signal 632. The phase hit detector 518 asserts the phase hit enable signal ph5_hit_en for one clock cycle.

Similarly, the second transition of the analog signal 632 causes the pulse 604 on the edge pulse signal 654 occurring on the fourth clock cycle. The phase hit detector circuit 547 identifies the phase signal ph0...ph7, phase signal ph6 in this case, most closely associated with the transition of the analog data signal 632 as indicated by the edge pulse signal 654. The phase hit detector 518 looks for adjacent phase out signals ph0_out...ph7_out where on signal is at a first level, e.g. 0, and the second is at a second level, e.g. 1. As a result, the phase hit detector 518 enables the phase hit enable signal ph6_hit_en corresponding to the phase signal ph6 as closest to the first transition of the analog signal 632. The phase hit detector 518 asserts the phase hit enable signal ph6_hit_en for one clock cycle. The PLL phase signal is then set based on some predetermined optimal phase offset between the transitions in the analog input data 242 (Figure 2) and the pixel clock 228 (Figure 2).

The phase hit detector 518 might alternatively assert the corresponding phase hit enable signal for longer than a clock cycle depending on whether the circuit 200 (Figure 2) detects a transition from a first level to a second level (e.g., 0 to 1 transition). For example, if the analog image data 242 (Figure 2) is changing from black to white every clock cycle then it is possible for the phase hit enable signal for a given phase to be active for all of the active pixels of a scan line.

A calibration circuit 242 calibrates the circuit 200 by determining the relationship between the measured phase hit and the desired optimal phase hit. The calibration circuit 242 receives the pixel clock 228 and the analog data signal 232 and provides the resulting signal to the edge detector circuit 240. By so manipulating the pixel clock 228 with the analog data signal 232, it is possible to determine the optimum phase alignment of the edge data of the

analog data signal 232. Optimum phase will result by positioning the edges of the analog data 232 so that they occur just after a rising edge, for example, of the pixel clock 228. By so calibrating the circuit 200, the analog data has the most settling time and will result in the least amount of digitized noise.

It should be readily apparent that one or more devices that include logic circuit might implement the present invention. A dedicated processor system that includes a microcontroller or a microprocessor might alternatively implement the present invention.

The invention additionally provides methods, which are described below. Moreover, the invention provides apparatus that performs or assists in performing the methods of the invention. This apparatus might be specially constructed for the required purposes or it might comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. The methods and algorithms presented herein are not necessarily inherently related to any particular computer or other apparatus. In particular, various general-purpose machines might be used with programs in accordance with the teachings herein or it might prove more convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these machines will appear from this description.

Useful machines or articles for performing the operations of the present invention include general-purpose digital computers or other similar devices. In all cases, there should be borne in mind the distinction between the method of operating a computer and the method of computation itself. The present invention relates also to method steps for operating a computer and for processing electrical or other physical signals to generate other desired physical signals.

The invention additionally provides a program and a method of operation of the program. The program is most advantageously implemented as a program for a computing machine, such as a general-purpose computer, a special purpose computer, a microprocessor, and the like.

The invention also provides a storage medium that has the program of the invention stored thereon. The storage medium is a computer-readable medium, such as a memory, and is read by the computing machine mentioned above.

A program is generally defined as a sequence of processes leading to a desired result. These processes, also known as instructions, are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of

electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated or processed. When stored, they might be stored in any computer-readable medium. It is convenient at times, principally for reasons of common usage, to refer to these signals as bits, data bits, samples, values, elements, symbols, characters, images, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are associated with the appropriate physical quantities, and that these terms are merely convenient labels applied to these physical quantities.

This detailed description is presented largely in terms of flowcharts, display images, algorithms, and symbolic representations of operations of data bits within a computer readable medium, such as a memory. Such descriptions and representations are the type of convenient labels used by those skilled in programming and/or the data processing arts to effectively convey the substance of their work to others skilled in the art. A person skilled in the art of programming might use this description to readily generate specific instructions for implementing a program according to the present invention. For the sake of economy, however, flowcharts used to describe methods of the invention are not repeated in this document for describing software according to the invention.

Often, for the sake of convenience only, it is preferred to implement and describe a program as various interconnected distinct software modules or features, collectively also known as software. This is not necessary, however, and there might be cases where modules are equivalently aggregated into a single program with unclear boundaries. In any event, the software modules or features of the present invention might be implemented by themselves, or in combination with others. Even though it is said that the program might be stored in a computer-readable medium, it should be clear to a person skilled in the art that it need not be a single memory, or even a single machine. Various portions, modules or features of it might reside in separate memories or separate machines where the memories or machines reside in the same or different geographic location. Where the memories or machines are in different geographic locations, they might be connected directly or through a network such as a local access network (LAN) or a global computer network like the Internet®.

In the present case, methods of the invention are implemented by machine operations. In other words, embodiments of the program of the invention are made such that they perform methods of the invention that are described in this document. These might be optionally performed in conjunction with one or more human operators performing some, but not all of them. As per the above, the users need not be collocated with each other, but each

only with a machine that houses a portion of the program. Alternately, some of these machines might operate automatically, without users and/or independently from each other.

Methods of the invention are now described. A person having ordinary skill in the art should recognize that the boxes described below might be implemented in different combinations, and in different order. Some methods might be used for determining a location of an object, some to determine an identity of an object, and some both.

Referring to Figure 7, after the circuit receives the analog data signal at box 710, it extracts the PLL clock from a reference signal, e.g., the horizontal synchronization signal, corresponding to the analog data signal. At box 712, the circuit generates a plurality of clock phases by delaying the PLL clock. The circuit identifies a transition of the analog signal at box 714 and then compares the transition with a predetermined threshold, e.g., a portion of the total signal swing (box 716). If the transition is, for example, smaller than the threshold, then circuit operation returns to box 714 where it identifies and tests the next transition.

If, however, the transition is greater than the threshold, the circuit generates an edge pulse at box 718. The circuit identifies one of the plurality of clock phases closest to the transition of the analog data as indicated by the rising edge of the edge pulse (box 720). At box 722, the circuit asserts a phase hit enable signal corresponding to the identified clock phase. A counter circuit keeps a count of asserted phase hit enable signals corresponding to each of the clock phases (box 724). The count continues for a predetermined time e.g., a predetermined number of image scan lines (box 726). If the predetermined time has not lapsed, then circuit operation returns to box 714 by identifying analog data signal transitions. If the predetermined time has lapsed, the circuit examines the count (box 728) and adjusts the frequency and phase of the pixel clock based on the count (box 730).

Figure 8 is a flowchart of an embodiment for a method of automatically adjusting the frequency of the pixel clock. Referring to Figure 8, the circuit examines the count for each clock phase over a predetermined time e.g., a predetermined number of scan lines. If some of the phase hit counts are very low (box 802), then the phase is set correctly (box 804). If the count for all or most of the phases greater than a predetermined number (box 802), the circuit adjusts the frequency by generating an appropriate frequency adjust signal that is provided to the PLL circuit (box 806). The ensure a properly adjusted frequency, the circuit clears the count (box 808) and enables the count once more (box 810) for a predetermined time e.g., a predetermined number of scan lines. The circuit operation returns to box 800 until the circuit verifies that the frequency of the pixel clock is properly adjusted.

The circuit might perform the above described frequency adjust operation many times in one vertical synchronization period since the PLL settling time after adjusting the frequency is usually only a few lines. The operation described above allows the circuit to measure the frequency of the pixel clock also in only a few lines e.g., in less than 100 lines. Thus, speed is a significant advantage to the circuit operation described above.

Figure 9 is a flowchart of an embodiment for a method of automatically adjusting the phase of the pixel clock. Referring to Figure 9, the circuit clears the count (box 900) and enables the count (box 902). After maintaining the count for a predetermined time (boxes 904 and 906), the circuit identifies the phase most closely associated with the transition of the analog data (box 908). At box 910, the circuit adjusts the phase of the pixel clock by generating an appropriate phase adjust signal that is provided to the phase adjust circuit.

After the circuit adjusts the frequency, only a few of the phase hit counters will show any appreciable number of hits (more than one counter will show hits due to the PLL jitter). The circuit might adjust the phase after only one read of the phase hit counters resulting in a substantial speed advantage over prior known techniques.

Having illustrated and described the principles of my invention in a preferred embodiment thereof, it should be readily apparent to those skilled in the art that the invention can be modified in arrangement and detail without departing from such principles. I claim all modifications coming within the spirit and scope of the accompanying claims.